INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2012; **00**:1–26
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme

An algebraic multigrid approach to solve XFEM based fracture problems

Axel Gerstenberger and Raymond S. Tuminaro*

1442 – Numerical Analysis and Applications, Sandia National Laboratories, P.O. Box 5800, MS 1320, Albuquerque, NM 87185-1320, USA, agerste@sandia.gov

SUMMARY

This article proposes an algebraic multigrid (AMG) approach to solve linear systems arising from applications where strong discontinuities are modeled by the eXtended Finite Element Method (XFEM). The application of AMG methods promises optimal scalability for solving large linear systems. However, the straight-forward (or 'black-box') use of existing AMG techniques for XFEM problems is often problematic. In this paper, we highlight the reasons for this behavior and propose a relatively simple adaptation that allows one to leverage existing AMG software mostly unchanged. Numerical tests demonstrate that optimal iterative convergence rates can be attained that are comparable to AMG convergence rates associated with linear systems for standard finite element approximations without discontinuities.

Received . . .

KEY WORDS: extended finite element method, algebraic multigrid, iterative solver, GMRES, conjugate gradient, parallel computing

1. INTRODUCTION

This article discusses algebraic multigrid methods (AMG) to solve linear systems arising from applications where the eXtended Finite Element Method (XFEM) is employed to capture strong discontinuities contained within the domain. The idea of the XFEM is to enrich the finite element space with discontinuous functions such that the discontinuity can be placed anywhere in the element and does not need to be aligned along element edges. In this way, a mesh with optimally shaped elements can be used for applications with strong discontinuities, where the position of the interface may not be known a priori. This flexibility, however, incurs a cost as the discrete system contains additional degrees of freedoms (DOFs) associated with the discontinuous functions, which requires adaptation of existing FEM software. XFEM examples for strong discontinuities can be found in computational fracture mechanics [1–6], the modeling of inclusion [7], multiphase-flow [8], or fluid-structure interaction [9–13], and for four dimensional space-time problems with moving interfaces [13, 14]. For a recent overview, see [15]. Our particular application is the simulation of brittle fracture, where the stress field near a crack tip is treated with an additional set of unknowns and approximation functions referred to as tip enrichment. The specific approximation functions differ depending on the specific cracking phenomena, e.g. brittle cracks or cohesive cracks.

[*] Correspondence to: 1442 – Numerical Analysis and Applications, Sandia National Laboratories, MS 9214, Livermore, CA 94551 USA, rstumin@sandia.gov

[†] The financial support of the Department of Energy under grant DE-SC0002137 and more generally the support from the DOE ASCR SciDAC ISICLES initiative are gratefully acknowledged.

While the discretization techniques have matured over the last decade, iterative solution algorithms for the resulting linear systems are not as developed. Iterative solution is generally hampered by the large condition number of the linear system that is primarily introduced by the tip enrichment. This tip enrichment problem has been considered in [16] within the context of one-level (i.e., not multigrid) iterative methods where special tip treatment was proposed to improve conditioning. Unfortunately, tip enrichment is not the only issue. This becomes apparent for large linear systems where one wishes to apply an optimal iterative technique such as AMG. It is well known that multigrid methods are often the most efficient schemes for very large linear systems [17, 18]. However, a 'black-box' application of existing AMG techniques for FEM to XFEM linear systems yields little improvement when compared to one-level iterative methods. The main reason is that AMG methods rely almost exclusively on the graph of the linear system. However, the matrix graphs of FEM and XFEM linear systems differ in how they represent strong discontinuities.

XFEM and geometric multigrid have been previously considered in [19]. A consistent level-set function is developed for use within different multigrid levels and different types of enrichment functions are treated separately within interpolation. Similarly, a local geometric multigrid method to resolve features near crack tips and generally small cracks has been proposed in [20]. A multilevel BPX preconditioner is examined in [21] for the generalized finite element method (GFEM) with simplicial grids. This approach makes use of an additional user-supplied matrix to guide the preconditioner construction. Standard smoothers are analyzed in [22] for a model GFEM problem. Line Gauss-Seidel schemes are then proposed for an algebraic multigrid method which rectify poor convergence issues on a model problem. An algebraic multigrid method for XFEM crack problems is proposed in [23]. It hinges on partitioning the linear system into a 2×2 block system based on whether unknowns are either near or far away from cracks. Almost optimal number of iterations have been achieved in [24]. However, the approach requires several modifications of the multigrid code and heavily uses XFEM information on all levels of the multigrid hierarchy, e.g. interface positions. It may therefore be seen as less optimal if one considers code modularity and code reuse. Additionally, special degrees-of-freedom associated with discontinuous basis functions are not coarsened but transferred unchanged to the coarsest level. This leads to many coarse unknowns and ultimately affects parallel scalability. The focus of our approach is to avoid XFEM specific information when constructing an AMG multigrid hierarchy. This allows one to employ an existing multigrid software library unchanged, and results in more modular code.

Two types of enrichment regions are present in our application: those corresponding to crack faces and those corresponding to crack tips. As we show, multigrid difficulties induced by crack faces are relatively easy to address. In fact, certain XFEM representations of crack faces do not introduce any difficulties for an AMG solver. Thus, the key is to understand which representations are AMG friendly and why. With this knowledge, we then revisit other representations to see how they can be suitably transformed so that they are more appropriate for AMG libraries. With such a transformation, we effectively adapt the input to the AMG library instead of changing the library. On the other hand, crack tip enrichments introduce severe ill-conditioning, but only a relatively few number of matrix rows are associated with this ill-conditioning. Hence, we treat tip enrichments as 'fine scale' features that are best dealt with on the finest level only and propose a simple multiplicative Schwarz decomposition that can be added as a fine level smoother.

The paper is outlined as follows: the governing equations for linear-elastic fracture mechanics along with commonly used XFEM techniques are reviewed in Section 2. In Section 3, the most widely used XFEM variants for modeling a crack surface are investigated in terms of the associated matrix graph with the aim of identifying aspects that might be problematic for an AMG solver. With this understanding, we then propose a transformation that removes these features. In Section 4, the overall solution approach is sketched. Here, both crack face and crack tip enrichments are treated. Computational performance is then verified in Section 5.

Note that we concentrate on steady problems in this paper as they are usually harder to solve than dynamic problems due to the lack of 'stabilizing' transient terms that usually improve iterative solver performance. However, the proposed algorithm applies equally well to the dynamic case.

DOI: 10.1002/nme

2. FRACTURE MECHANICS USING THE EXTENDED FINITE ELEMENT METHOD

The strong form for a static, linear elastic fracture problem is given as

$$
\begin{aligned}
-\boldsymbol{\nabla} \cdot \underset{\sim}{\boldsymbol{\sigma}} - \boldsymbol{f} &= 0 \quad \text{in } \varOmega &(1)\\
\boldsymbol{u} - \boldsymbol{u}^{\mathrm{D}} &= 0 \quad \text{in } \varGamma^{\mathrm{D}} &(2)\\
\underset{\sim}{\boldsymbol{\sigma}} \cdot \boldsymbol{n} - \boldsymbol{t}^{\mathrm{N}} &= 0 \quad \text{in } \varGamma^{\mathrm{N}} &(3)\\
\underset{\sim}{\boldsymbol{\sigma}} \cdot \boldsymbol{n} &= 0 \quad \text{in } \varGamma^{\mathrm{crack}} \;\; . &(4)
\end{aligned}
$$

Here, $\underset{\sim}{\boldsymbol{\sigma}}$ denotes the Cauchy stress, $\boldsymbol{f}$ a volume force, and $\boldsymbol{u}$ the displacement in domain $\varOmega$. Along the Dirichlet boundary $\varGamma^{\mathrm{D}}$, $\boldsymbol{u}^{\mathrm{D}}$ denotes the given displacement and $\boldsymbol{t}^{\mathrm{N}}$ denotes the surface traction along the Neumann boundary $\varGamma^{\mathrm{N}}$. Along all crack faces $\varGamma^{\mathrm{crack}}$, we assume traction-free Neumann conditions. The corresponding weak form after integration by parts is given as

$$
0 = \int_{\varOmega} \boldsymbol{\nabla} \delta \boldsymbol{d} : \underset{\sim}{\boldsymbol{\sigma}} \mathrm{d}\boldsymbol{x} - \int_{\varOmega} \delta \boldsymbol{d} \cdot \boldsymbol{f} \mathrm{d}\boldsymbol{x} - \int_{\varGamma^{\mathrm{N}}} \delta \boldsymbol{d} \cdot \boldsymbol{t}^{\mathrm{N}} \mathrm{d}\boldsymbol{x} \;\; . \tag{5}
$$

The displacement is discretized as follows

$$
\boldsymbol{u}^{\mathrm{h}}(\boldsymbol{x}) = \sum_{I} N_I(\boldsymbol{x}) \boldsymbol{u}_I + \sum_{K} N_K(\boldsymbol{x}) \psi^{\mathrm{s}}_K(\boldsymbol{x}) \boldsymbol{u}^{\mathrm{s}}_K + \sum_{L} N_L(\boldsymbol{x}) \sum_{J=1}^{n^{\mathrm{tip}}} \psi^{\mathrm{tip}}_{LJ}(\boldsymbol{x}) \boldsymbol{u}^{\mathrm{tip}}_{LJ} \tag{6}
$$

The different summations correspond to normal, jump-enriched, and tip-enriched nodes. The standard shape functions are denoted by $N(\boldsymbol{x})$ and the enrichment function for crack faces and tips are denoted by $\psi^{\mathrm{s}}_K(\boldsymbol{x})$ and $\psi^{\mathrm{tip}}_{LJ}(\boldsymbol{x})$, respectively. For our numerical examples, the so-called *shifted enrichment* [25] is employed, where $\psi^{\mathrm{s}}_K(\boldsymbol{x})$ is defined as

$$
\psi^{\mathrm{s}}_K(\boldsymbol{x}) = \frac{1}{2} \left(H(\boldsymbol{x}) - H(\boldsymbol{x}_K) \right) , \tag{7}
$$

with $H(\boldsymbol{x})$ denoting the Heaviside function given as

$$
H(\boldsymbol{x}) = \begin{cases} +1 & \text{in } \varOmega^{\mathrm{a}} \\ -1 & \text{in } \varOmega^{\mathrm{b}} \;\; . \end{cases} \tag{8}
$$

The shifted tip enrichment functions are defined as

$$
\psi^{\mathrm{tip}}_{LJ}(\boldsymbol{x}) = \frac{1}{2} \left(F_J(\boldsymbol{x}) - F_J(\boldsymbol{x}_L) \right) \tag{9}
$$

where the four tip enrichment functions $F_J(\boldsymbol{x})$ are

$$
\begin{aligned}
F_1(r,\theta) &= \sqrt{r} \sin\left(\frac{\theta}{2}\right), & F_2(r,\theta) &= \sqrt{r} \cos\left(\frac{\theta}{2}\right), &(10)\\
F_3(r,\theta) &= \sqrt{r} \sin\left(\frac{\theta}{2}\right) \sin(\theta) , & F_4(r,\theta) &= \sqrt{r} \cos\left(\frac{\theta}{2}\right) \sin(\theta) \;\; . &(11)
\end{aligned}
$$

Other variants for crack and tip enrichment functions can be found in the literature, see e.g. [15] for an overview. The position of the crack face and the crack tips is often defined by level-set functions [7, 25]. As the solution methods proposed in this paper are independent of these, they are not discussed further.

The linear system is constructed by assembling element stiffness matrices and vectors into global matrix $\boldsymbol{K}$ and $\boldsymbol{f}$ as

$$
\boldsymbol{K} = \mathop{\mathsf{A}}_{e} \left[\int_{\varOmega_e} \frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{x}} \boldsymbol{C} \frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{x}} \mathrm{d}\boldsymbol{x} \right] \tag{12}
$$

and

$$\boldsymbol{f}^{\mathrm{ext}} = \mathop{\mathrm{A}}_{e}\left[\int_{\Omega_e} \boldsymbol{\phi}\,\boldsymbol{f}^{\mathrm{h}}\mathrm{d}\boldsymbol{x} + \int_{\Gamma_e^{\mathrm{N}}} \boldsymbol{\phi}\,\boldsymbol{t}^{\mathrm{N,h}}\mathrm{d}\boldsymbol{x}\right] \tag{13}$$

where $\boldsymbol{\phi}$ is a vector of all enriched and unenriched approximation functions. For transient problems, the mass matrix is required, which is computed and assembled as

$$\boldsymbol{M} = \mathop{\mathrm{A}}_{e}\left[\int_{\Omega_e} \boldsymbol{\phi}\rho\boldsymbol{\phi}\mathrm{d}\boldsymbol{x}\right] \quad . \tag{14}$$

The density is denoted by ρ. The resulting linear system is of the form

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{b} \tag{15}$$

where the operator $\boldsymbol{A}$ equals $\boldsymbol{K}$ in our case of static, linear elastic fracture mechanics, but could also contain transient terms or could be the effective tangent stiffness matrix of a non-linear XFEM problem. The remainder of the paper addresses the solution of Eq. (15).

3. HEAVISIDE ENRICHMENT STRATEGIES AND THEIR EFFECT ON THE LINEAR SYSTEM GRAPH

3.1. Linear system graphs

Mesh information is not generally supplied or used by an AMG algorithm. Instead, a graph is constructed by examining matrix entries. This graph effectively occupies the role of the mesh in traditional geometric multigrid. Ultimately, the graph determines the support of interpolation basis functions and more generally how operators are coarsened. These decisions are based on an important assumption that the matrix graph represents strong influences. That is, a large change in the solution at the location associated with one vertex should be accompanied by commensurate large solution changes at locations associated with neighboring vertices.

Given its importance, we now investigate matrix graphs. The primary motivation is to highlight the different graphs associated with different enriched FE representations and how only some of these satisfy the strong influence assumption. We begin by defining a matrix graph $\mathcal{G}$. Formally, it is given by a set of vertices $\mathcal{V}$ and a set of edges $\mathcal{E}$

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}). \tag{16}$$

A single vertex represents unknown degrees-of-freedom at a single mesh node. It is typically assumed that degrees-of-freedom at a given mesh node are ordered consecutively, e.g. $[u_{1,x}, u_{1,y}, u_{2,x}, u_{2,y}, \ldots]$. In this setting, the k^{th} vertex represents degrees-of-freedom numbered from $(k-1)\alpha + 1$ to $k\alpha$ where α is the number of degrees-of-freedom per node. The edge set consists of vertex pairs

$$\mathcal{E} = \{(\mathcal{V}_k, \mathcal{V}_\ell) \mid \exists A_{ij} \neq 0 \;\text{ where }\; i \in \mathcal{V}_k,\; j \in \mathcal{V}_\ell\}. \tag{17}$$

Thus, vertices correspond to matrix rows and edges correspond to nonzeros for scalar PDEs.[†] For PDE systems, block matrix rows define vertices while nonzero sub-matrix blocks define edges.

3.2. Discretization of a 1D truss with a 'crack'

3.2.1. Finite Elements Consider a linear elastic material in a one dimensional truss as depicted in Figure 1. Four standard elements (see e.g. [26]) can be used to model the crack: two to the left of the crack and two to the right of the crack. To enable the discontinuity, separate FE nodes are introduced

[†] It should be noted that often the magnitude of A_{ij} is compared with a small threshold instead of simply checking whether it is nonzero.

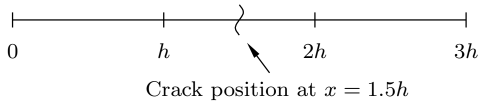
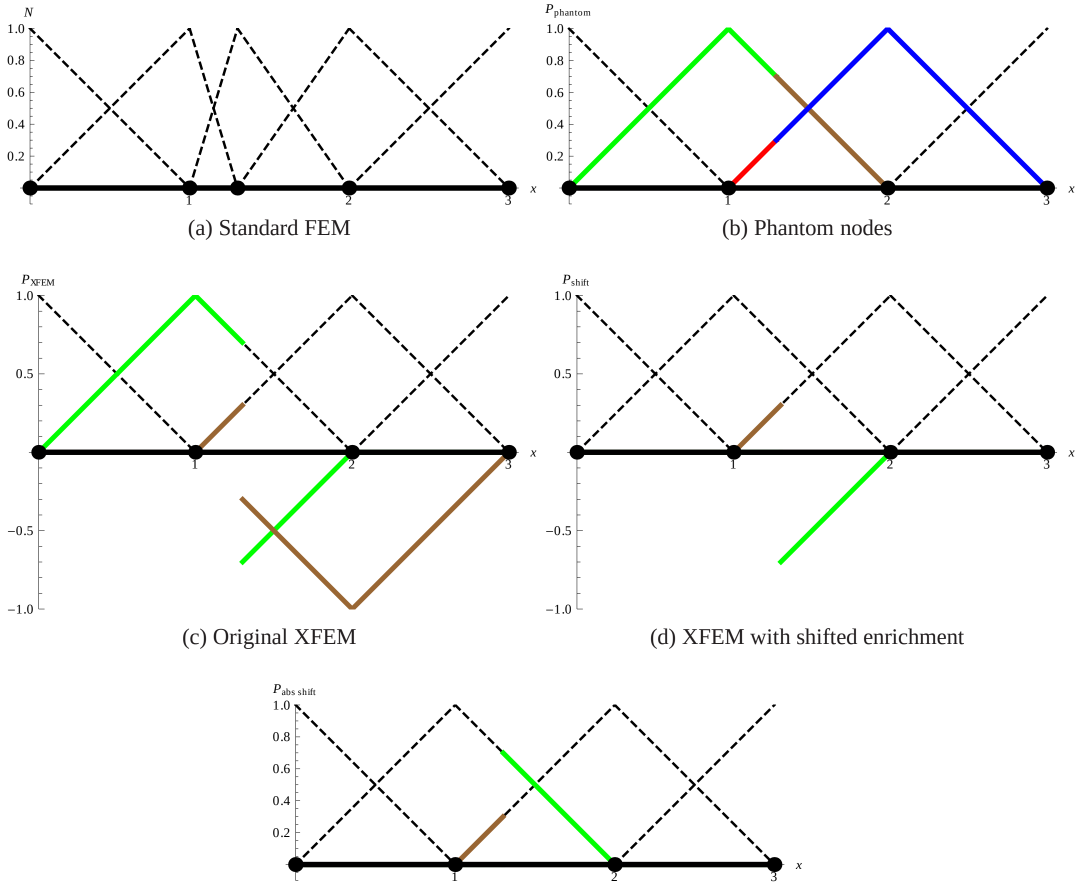


Figure 1. Setup of a 1d truss example. The truss is cracked in the middle at $x = 1.5h$.



(a) Standard FEM

(b) Phantom nodes

(c) Original XFEM

(d) XFEM with shifted enrichment

(e) XFEM with absolute shifted enrichment

Figure 2. FE and enriched FE shape functions. Dashed lines correspond to standard FE shape functions, while full lines indicate enriched shape functions. For better illustration of the discont. approximation functions, the crack is positioned at $x_{\text{crack}} = 1.3h$ in these pictures.

such that the two elements adjacent to the crack are not connected to each other. The shape functions for each node are depicted in Figure 2a and a global stiffness matrix and mass matrix is given in Table Ia corresponding to element widths of $h/2$ for crack-adjacent elements and h for elements far from the crack. Notice that the matrix graph splits into two disjoint pieces corresponding to the left and right subdomains. That is, the matrix graph accurately reflects the disconnected nature of the two subdomains.

3.2.2. Phantom nodes The phantom node approach is an XFEM variant with discontinuous shape function that has been proposed by several authors, for instance [2, 5, 7, 27], under different names and has been shown to be identical to the original XFEM in [28]. The common philosophy is to model the domains on each side of the crack separately with a discontinuity in the element containing the crack and then placing both discrete domains into the same global system.

Table I. Stiffness and mass matrices for disconnected 1d truss.

Method	$\boldsymbol{K}/(\frac{EA}{2h})$	$\boldsymbol{M}/(\frac{1}{4}\frac{\rho Ah}{6})$
a) Standard	$\begin{bmatrix} 2 & -2 & 0 & 0 & 0 & 0 \\ -2 & 6 & -4 & 0 & 0 & 0 \\ 0 & -4 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & -4 & 0 \\ 0 & 0 & 0 & -4 & 6 & -2 \\ 0 & 0 & 0 & 0 & -2 & 2 \end{bmatrix}$	$\begin{bmatrix} 8 & 4 & 0 & 0 & 0 & 0 \\ 4 & 12 & 2 & 0 & 0 & 0 \\ 0 & 2 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 2 & 0 \\ 0 & 0 & 0 & 2 & 12 & 4 \\ 0 & 0 & 0 & 0 & 4 & 8 \end{bmatrix}$
b) Phantom	$\begin{bmatrix} 2 & -2 & 0 & 0 & 0 & 0 \\ -2 & 3 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 3 & -2 \\ 0 & 0 & 0 & 0 & -2 & 2 \end{bmatrix}$	$\begin{bmatrix} 8 & 4 & 0 & 0 & 0 & 0 \\ 4 & 15 & 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 2 & 0 \\ 0 & 2 & 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 15 & 4 \\ 0 & 0 & 0 & 0 & 4 & 8 \end{bmatrix}$
c) Original XFEM	$\begin{bmatrix} 2 & -2 & -2 & 0 & 0 & 0 \\ -2 & 4 & 2 & -2 & 0 & 0 \\ -2 & 2 & 4 & 0 & -2 & 0 \\ 0 & -2 & 0 & 4 & -2 & -2 \\ 0 & 0 & -2 & -2 & 4 & 2 \\ 0 & 0 & 0 & -2 & 2 & 2 \end{bmatrix}$	$\begin{bmatrix} 8 & 4 & 4 & 0 & 0 & 0 \\ 4 & 16 & 14 & 4 & 0 & 0 \\ 4 & 14 & 16 & 0 & 4 & 0 \\ 0 & 4 & 0 & 16 & -14 & 4 \\ 0 & 0 & 4 & -14 & 16 & -4 \\ 0 & 0 & 0 & 4 & -4 & 8 \end{bmatrix}$
d) Shifted XFEM	$\begin{bmatrix} 2 & -2 & 0 & 0 & 0 & 0 \\ -2 & 4 & -1 & -2 & -1 & 0 \\ 0 & -1 & 1 & 1 & 0 & 0 \\ 0 & -2 & 1 & 4 & 1 & -2 \\ 0 & -1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & -2 & 0 & 2 \end{bmatrix}$	$\begin{bmatrix} 8 & 4 & 0 & 0 & 0 & 0 \\ 4 & 16 & -1 & 4 & 4 & 0 \\ 0 & -1 & 1 & -2 & 0 & 0 \\ 0 & 4 & -2 & 16 & 1 & 4 \\ 0 & 4 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 4 & 0 & 8 \end{bmatrix}$
e) Abs. Shifted XFEM	$\begin{bmatrix} 2 & -2 & 0 & 0 & 0 & 0 \\ -2 & 4 & 1 & -2 & -1 & 0 \\ 0 & 1 & 1 & -1 & 0 & 0 \\ 0 & -2 & -1 & 4 & 1 & -2 \\ 0 & -1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & -2 & 0 & 2 \end{bmatrix}$	$\begin{bmatrix} 8 & 4 & 0 & 0 & 0 & 0 \\ 4 & 16 & 1 & 4 & 2 & 0 \\ 0 & 1 & 1 & 2 & 0 & 0 \\ 0 & 4 & 2 & 16 & 1 & 4 \\ 0 & 2 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 4 & 0 & 8 \end{bmatrix}$

Specifically, the displacement is discretized as

$$u^{\mathrm{p}} = \sum_{I\in\{1,4\}} N_I(x)u_i + \sum_{I\in\{2,3\}} N_I(x)\psi^{\mathrm{a}}(x)u_I^{\mathrm{a}} + \sum_{I\in\{2,3\}} N_I(x)\psi^{\mathrm{b}}(x)u_I^{\mathrm{b}} \tag{18}$$

with

$$\psi^{\mathrm{a}}(x) = \begin{cases} 1 & \text{in } \Omega^{\mathrm{a}} \\ 0 & \text{in } \Omega^{\mathrm{b}} \end{cases} \quad \text{and} \quad \psi^{\mathrm{b}}(x) = \begin{cases} 0 & \text{in } \Omega^{\mathrm{a}} \\ 1 & \text{in } \Omega^{\mathrm{b}} \end{cases} \tag{19}$$

where Ω^{a} and Ω^{b} refer to the left and right domains, respectively. The resulting shape functions in Figure 2b demonstrate that one can group the unknowns into unknowns belonging to the left and to the right domain as

$$\boldsymbol{u}^{\mathrm{p}} = \begin{bmatrix} u_1 & u_2^{\mathrm{a}} & u_2^{\mathrm{b}} & u_3^{\mathrm{a}} & u_3^{\mathrm{b}} & u_4 \end{bmatrix}^{\mathrm{T}} \tag{20}$$

where u_1, u_2^{a}, and u_3^{a} influence the displacement only in domain Ω^{a} and the remaining DOFs influence only Ω^{b}. The existence of unknowns at nodes beyond the interface is the main reason for terms like *overlapping elements* [27] or *phantom node approach* [5]. The matrices in Table Ib are based on the unknown ordering given in Eq. (20). As with standard finite elements, the associated matrix graph consists of two disconnected pieces accurately reflecting the lack of influence of the left domain on the right domain (and vice versa).

3.2.3. Original XFEM The original XFEM discretization of the problem, which follows [1, 3], uses three connected elements. The displacement is approximated by

$$u^{\mathrm{x}} = \sum_{I=1}^{4} N_I(x)u_i + \sum_{I\in\{2,3\}} N_I(x)\psi^{\mathrm{x}}(x)u_I^{\mathrm{x}}. \tag{21}$$

where $\psi^{\mathrm{x}}(x)$ is the Heaviside function given as

$$\psi^{\mathrm{x}}(x) = H(x) = \begin{cases} +1 & \text{in } \Omega^{\mathrm{a}} \\ -1 & \text{in } \Omega^{\mathrm{b}} \end{cases} . \tag{22}$$

The setup can be described as a continuous domain (using four standard FE approximations) and two additional enriched approximations. The resulting shape function for each unknown are given in Figure 2c and the global stiffness matrix and mass matrix are given in Table Ic. For the latter, the unknown ordering in the discrete displacement vector $\boldsymbol{u}^{\mathrm{x}}$ is

$$\boldsymbol{u}^{\mathrm{x}} = \begin{bmatrix} u_1 & u_2 & u_2^{\mathrm{x}} & u_3 & u_3^{\mathrm{x}} & u_4 \end{bmatrix}^{\mathrm{T}} . \tag{23}$$

Unlike the two previous formulations, matrix connections coupling the left and right sub-domains are now present.

3.2.4. Shifted enrichment A variant of the original XFEM discretization is the so called shifted enrichment as introduced in [25], which is popular because it recovers the Kronecker delta property from standard finite element methods for the unenriched DOFs. Here the displacement is approximated by

$$u^{\mathrm{s}} = \sum_{I=1}^{4} N_I(x)u_I + \sum_{I\in\{2,3\}} N_I(x)\psi_I^{\mathrm{s}}(x)u_I^{\mathrm{s}}. \tag{24}$$

and the enrichment function $\psi_I^{\mathrm{s}}(\boldsymbol{x})$ for node I is given as

$$\psi_I^{\mathrm{s}}(\boldsymbol{x}) = \frac{1}{2}\left(\psi^{\mathrm{x}}(\boldsymbol{x}) - \psi^{\mathrm{x}}(\boldsymbol{x}_I)\right). \tag{25}$$

The unknown ordering for the matrices in Table Id is

$$\boldsymbol{u}^{\mathrm{s}} = \begin{bmatrix} u_1 & u_2 & u_2^{\mathrm{s}} & u_3 & u_3^{\mathrm{s}} & u_4 \end{bmatrix}^{\mathrm{T}} \tag{26}$$

As a result, two unenriched shape function cross the crack and the two enriched shape functions are zero at all nodes (see Figure 2d). Again, non-zero matrix connections between left and right sub-domains are present.

3.2.5. Absolute shifted enrichment For the discussion in subsequent chapters, we propose a variant of shifted enrichment which takes the absolute value of $\psi_I^{\mathrm{s}}(\boldsymbol{x})$:

$$\psi_I^{|\mathrm{s}|}(\boldsymbol{x}) = \frac{1}{2}\left|\psi^{\mathrm{x}}(\boldsymbol{x}) - \psi^{\mathrm{x}}(\boldsymbol{x}_I)\right| \tag{27}$$

and uses this in Eq. (24) as a replacement for $\psi_I^{\mathrm{s}}(\boldsymbol{x})$ This leads to strictly non-negative approximation functions for linear Lagrange polynomials and so the corresponding mass matrix has only positive entries. The difference with shifted enrichment can be seen by comparing Figure 2d with Figure 2e.

DOI: 10.1002/nme

The unknown ordering for the matrices in Table Ie is

$$\boldsymbol{u}^{|\mathrm{s}|} = \begin{bmatrix} u_1 & u_2 & u_2^{|\mathrm{s}|} & u_3 & u_3^{|\mathrm{s}|} & u_4 \end{bmatrix}^{\mathrm{T}} \tag{28}$$

where $u_I^{|\mathrm{s}|}$ replaces u_I^{s} in Eq. (24). As in the shifted enrichment, non-zero matrix connections between left and right sub-domains are present.

3.3. Change of basis transformation

The previous sub-section demonstrates that the original XFEM enrichment as well as the shifted enrichment XFEM yield matrices which *hide* the disconnected nature of the sub-domains as opposed to the phantom node approach and standard FEM where matrices clearly express this disconnection. The failure to properly express disconnections complicates the AMG process as described in Section 4. Rather than rewrite AMG procedures or an XFEM implementation, we propose a change of basis transformation

$$(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{G})(\boldsymbol{G}^{-1}\boldsymbol{u}) = \boldsymbol{G}^{\mathrm{T}}\boldsymbol{b} \tag{29}$$

where $\boldsymbol{G}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{G}$, $\boldsymbol{G}^{-1}\boldsymbol{u}$, and $\boldsymbol{G}^{\mathrm{T}}\boldsymbol{b}$ describe a new matrix, new unknown vector, and new right hand side respectively. The basic idea is to change a matrix which hides disconnections to an equivalent one where they are respected. In this way standard AMG procedures can be employed on the transformed matrix without making large software changes on either the solver or discretization side. Codes can simply apply the transformation before and after the solution of the linear system. Hence, software with respect to boundary conditions, visualization and similar things, that takes advantage of the specific properties of the original XFEM or the shifted enrichment XFEM remains untouched. Further, it will not be necessary to explicitly store transformation matrices due to their inherent simplicity as will be illustrated later in this section.

To find a transformation, we rely on an equivalence that exists between different XFEM representations. In particular, the phantom node approach has been shown to be mathematically identical to the original XFEM formulation [29] which means that there exists a transformation matrix $\boldsymbol{G}_{\mathrm{p}}^{\mathrm{x}}$ between these representations. In the following, we repeat the argument in an adapted form and show that there exists a similar transformation between the shifted and absolute shifted enrichment approach and the phantom node approach.

Note that we do not attempt to transform between an XFEM and a corresponding FEM representation, because creating the transformation matrix would require a new FEM mesh near the crack and the evaluation of integrals over element subsection. As we show, transformation between XFEM representation do not require the evaluation of integrals and are therefore preferred.

3.3.1. Transformation between original XFEM and phantom node approach Consider a matrix $\boldsymbol{G}_{\mathrm{p}}^{\mathrm{x}}$ such that

$$u^{\mathrm{x}} = \boldsymbol{G}_{\mathrm{p}}^{\mathrm{x}} u^{\mathrm{p}}. \tag{30}$$

Recalling Eq. (19) and Eq. (22) and recognizing their equivalence implies that in an intersected element

$$\sum_{I\in C} N_I(\boldsymbol{x})u_I + N_I(\boldsymbol{x})\psi^{\mathrm{x}}(\boldsymbol{x})u_I^{\mathrm{x}} = \sum_{I\in C} N_I(\boldsymbol{x})\psi^{\mathrm{a}}(\boldsymbol{x})u_I^{\mathrm{a}} + N_I(\boldsymbol{x})\psi^{\mathrm{b}}(\boldsymbol{x})u_I^{\mathrm{b}} \tag{31}$$

where the set C denotes nodes which are adjacent to the crack (i.e., nodes which have been enriched). In our simple example, C would consist of the 2nd and 3rd nodes. Eq. (31) is obtained by equating Eq. (19) with Eq. (22) and then eliminating unenriched nodes (i.e., the 1st and 4th nodes in our example) from both sides of the equation. We also have the following two relationships

$$\psi^{\mathrm{a}}(x) = \frac{1+\psi^{\mathrm{x}}(x)}{2} \quad \text{and} \quad \psi^{\mathrm{b}}(x) = \frac{1-\psi^{\mathrm{x}}(x)}{2}. \tag{32}$$

Thus, Eq. (31) can be re-written as

$$\sum_{I\in C} N_I(\boldsymbol{x})\left(u_I + \psi^{\mathrm{x}}(\boldsymbol{x})u_I^{\mathrm{x}}\right) = \sum_{I\in C} N_I(\boldsymbol{x})\left(\frac{1+\psi^{\mathrm{x}}(\boldsymbol{x})}{2}u_I^{\mathrm{a}} + \frac{1-\psi^{\mathrm{x}}(\boldsymbol{x})}{2}u_I^{\mathrm{b}}\right). \tag{33}$$

For arbitrary shape functions and enrichment functions, we can conclude that the two summations are equal if

$$u_I = \frac{u_I^{\mathrm{a}} + u_I^{\mathrm{b}}}{2} \quad \text{and} \quad u_I^{\mathrm{x}} = \frac{u_I^{\mathrm{a}} - u_I^{\mathrm{b}}}{2}. \tag{34}$$

In matrix form, this can be written for each node $I \in C$ as

$$\begin{bmatrix} u_I \\ u_I^{\mathrm{x}} \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix} \begin{bmatrix} u_I^{\mathrm{a}} \\ u_I^{\mathrm{b}} \end{bmatrix}. \tag{35}$$

Applied to the 3 element example, $\boldsymbol{G}_{\mathrm{p}}^{\mathrm{x}}$ is defined by

$$\begin{bmatrix} u_1 \\ u_2 \\ u_2^{\mathrm{x}} \\ u_3 \\ u_3^{\mathrm{x}} \\ u_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & -0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 & -0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2^{\mathrm{a}} \\ u_2^{\mathrm{b}} \\ u_3^{\mathrm{a}} \\ u_3^{\mathrm{b}} \\ u_4 \end{bmatrix} . \tag{36}$$

It is block diagonal where the 2 × 2 blocks correspond to nodal functions which are modified by ψ^{x}, ψ^{a}, or ψ^{b}. Each of these 2 × 2 blocks has the same entries as the 2 × 2 diagonal blocks in Eq. (35).

It is important to notice that the above transformation did not rely on any property of $N_I(\boldsymbol{x})$ nor is there any assumption about the number of nodes in C. The transformation only relies on the relations given in Eq. (32). These relations, however, are unaffected by the use of higher order shape functions and further they are also valid in higher dimensions. For example in two dimensions,

$$\psi^{\mathrm{a}}(x,y) = \begin{cases} 1 & \text{in } \Omega^{\mathrm{a}} \\ 0 & \text{in } \Omega^{\mathrm{b}} \end{cases}, \quad \psi^{\mathrm{b}}(x,y) = \begin{cases} 0 & \text{in } \Omega^{\mathrm{a}} \\ 1 & \text{in } \Omega^{\mathrm{b}} \end{cases} \quad \text{and} \quad \psi^{\mathrm{x}}(x,y) = \begin{cases} +1 & \text{in } \Omega^{\mathrm{a}} \\ -1 & \text{in } \Omega^{\mathrm{b}} \end{cases} \tag{37}$$

where Ω^{a} and Ω^{b} define two sub-domains on opposites sides of a crack. Thus, it follows that the same basic 2 × 2 transformation matrix is valid when original XFEM and phantom node discretizations are employed in higher dimensions and with higher order discretizations. Translating between these two representations simply requires applying the same 2 × 2 matrix to unknowns associated with each enrichment function and its corresponding unenriched counterpart. In the case of vector fields, e.g. displacement field $\boldsymbol{u}(\boldsymbol{x})$, each of the vector field components is discretized independently, hence, the transformation is applied to each discretized vector field component independently. An example for such transformation is given at the end of this section.

3.3.2. Transformation between shifted enrichment and phantom node approach A similar exercise can be performed for shifted enrichment representations. Recalling Eq. (18) and Eq. (24) and recognizing their equivalence implies that for all enriched nodes $I \in C$

$$\sum_{I\in C} N_I(\boldsymbol{x})u_I + N_I(\boldsymbol{x})\psi_I^{\mathrm{s}}(\boldsymbol{x})u_I^{\mathrm{s}} = \sum_{I\in C} N_I(\boldsymbol{x})\psi^{\mathrm{a}}(\boldsymbol{x})u_I^{\mathrm{a}} + N_I(\boldsymbol{x})\psi^{\mathrm{b}}(\boldsymbol{x})u_I^{\mathrm{b}}. \tag{38}$$

Using Eq. (25) and Eq. (19), Eq. (38) can be re-written as

$$\sum_{I\in C} N_I(\boldsymbol{x})\left(u_I + \frac{1}{2}\left(\psi^{\mathrm{x}}(\boldsymbol{x}) - \psi^{\mathrm{x}}(\boldsymbol{x}_I)\right)u_I^{\mathrm{s}}\right) = \sum_{I\in C} N_I(\boldsymbol{x})\left(\frac{1+\psi^{\mathrm{x}}}{2}u_I^{\mathrm{a}} + \frac{1-\psi^{\mathrm{x}}}{2}u_I^{\mathrm{b}}\right). \tag{39}$$

For arbitrary shape functions, this equation holds, if

$$u_I + \frac{1}{2}\left(\psi^{\mathrm{x}}(\boldsymbol{x}) - \psi^{\mathrm{x}}(\boldsymbol{x}_I)\right)u_I^{\mathrm{s}} = \frac{1}{2}(u_I^{\mathrm{a}} + u_I^{\mathrm{b}} + \psi^{\mathrm{x}}(\boldsymbol{x})(u_I^{\mathrm{a}} - u_I^{\mathrm{b}})). \tag{40}$$

Evaluating Eq. (40) at two arbitrary points on opposite sides of the crack gives

$$u_I + \frac{1}{2}\left(1 - \psi^{\mathrm{x}}(\boldsymbol{x}_I)\right) u_I^{\mathrm{s}} = \frac{1}{2}(u_I^{\mathrm{a}} + u_I^{\mathrm{b}} + (u_I^{\mathrm{a}} - u_I^{\mathrm{b}})) \tag{41}$$

$$u_I + \frac{1}{2}\left(-1 - \psi^{\mathrm{x}}(\boldsymbol{x}_I)\right) u_I^{\mathrm{s}} = \frac{1}{2}(u_I^{\mathrm{a}} + u_I^{\mathrm{b}} - (u_I^{\mathrm{a}} - u_I^{\mathrm{b}})). \tag{42}$$

After some algebraic manipulations, we obtain

$$u_I^{\mathrm{a}} = u_I - \frac{1}{2}\left(\psi^{\mathrm{x}}(\boldsymbol{x}_I) - 1\right) u_I^{\mathrm{s}} \;\text{ and }\; u_I^{\mathrm{b}} = u_I - \frac{1}{2}\left(\psi^{\mathrm{x}}(\boldsymbol{x}_I) + 1\right) u_I^{\mathrm{s}} \;. \tag{43}$$

Solving for u_I and u_I^{s} and writing the result in matrix form, we obtain

$$\begin{bmatrix} u_I \\ u_I^{\mathrm{s}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\psi^{\mathrm{x}}(\boldsymbol{x}_I) + \frac{1}{2} & -\frac{1}{2}\psi^{\mathrm{x}}(\boldsymbol{x}_I) + \frac{1}{2} \\ 1 & -1 \end{bmatrix} \begin{bmatrix} u_I^{\mathrm{a}} \\ u_I^{\mathrm{b}} \end{bmatrix} \;. \tag{44}$$

Applied to the 3-element example, the transformation matrix is

$$\begin{bmatrix} u_1 \\ u_2 \\ u_2^{\mathrm{s}} \\ u_3 \\ u_3^{\mathrm{s}} \\ u_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2^{\mathrm{a}} \\ u_2^{\mathrm{b}} \\ u_3^{\mathrm{a}} \\ u_3^{\mathrm{b}} \\ u_4 \end{bmatrix} \;. \tag{45}$$

Note that the small 2×2 sub-matrix is different depending on where a node is located relative to the crack and so node locations relative to each crack must be evaluated when constructing this transformation.

3.3.3. Transformation between absolute shifted enrichment and phantom node approach The equivalent to Eq. (40) for this enrichment is

$$u_I + \frac{1}{2}\left|\psi^{\mathrm{x}}(\boldsymbol{x}) - \psi^{\mathrm{x}}(x_I)\right| u_I^{|\mathrm{s}|} = \frac{1}{2}(u_I^{\mathrm{a}} + u_I^{\mathrm{b}} + \psi^{\mathrm{x}}(u_I^{\mathrm{a}} - u_I^{\mathrm{b}})). \tag{46}$$

Evaluating at two arbitrary points on opposite crack sides gives

$$u_I + \frac{1}{2}\left|1 - \psi^{\mathrm{x}}(x_I)\right| u_I^{|\mathrm{s}|} = u_I^{\mathrm{a}} \tag{47}$$

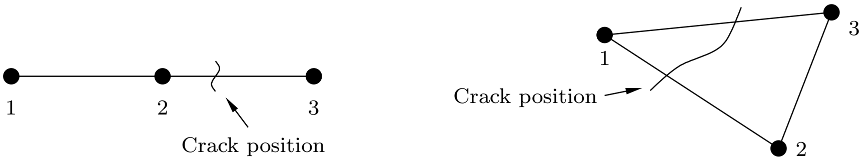
$$u_I + \frac{1}{2}\left|-1 - \psi^{\mathrm{x}}(x_I)\right| u_I^{|\mathrm{s}|} = u_I^{\mathrm{b}}. \tag{48}$$

Solving for u_I and $u_I^{|\mathrm{s}|}$ and writing the result in matrix form leads to

$$\begin{bmatrix} u_I \\ u_I^{|\mathrm{s}|} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} + \frac{1}{2}\psi^{\mathrm{x}}(x_I) & \frac{1}{2} - \frac{1}{2}\psi^{\mathrm{x}}(x_I) \\ -\psi^{\mathrm{x}}(x_I) & \psi^{\mathrm{x}}(x_I) \end{bmatrix} \begin{bmatrix} u_I^{\mathrm{a}} \\ u_I^{\mathrm{b}} \end{bmatrix} \;. \tag{49}$$

Applied to the 3-element example, $\boldsymbol{G}_{\mathrm{p}}^{|\mathrm{s}|}$ is defined by

$$\begin{bmatrix} u_1 \\ u_2 \\ u_2^{|\mathrm{s}|} \\ u_3 \\ u_3^{|\mathrm{s}|} \\ u_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2^{\mathrm{a}} \\ u_2^{\mathrm{b}} \\ u_3^{\mathrm{a}} \\ u_3^{\mathrm{b}} \\ u_4 \end{bmatrix} \tag{50}$$



(a) 1d quadratic truss element

(b) 2d linear displacement element

Figure 3. Illustrative examples.

which can be re-ordered as

$$\begin{bmatrix} u_1 \\ u_2 \\ u_2^{|s|} \\ u_3 \\ u_3^{|s|} \\ u_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2^{\mathrm{a}} \\ u_2^{\mathrm{b}} \\ u_3^{\mathrm{b}} \\ u_3^{\mathrm{a}} \\ u_4 \end{bmatrix} . \tag{51}$$

In Eq. (51), we have a repeating pattern of 2 × 2 sub-matrices with 1 on the diagonal and -1 as the lower left matrix entry at each node, while in Eq. (50), the sub-matrix pattern depends on the nodal position. In other words, we can choose to have a fixed transformation matrix with a nodal position dependent phantom node ordering (Eq. (51)) or vice versa (Eq. (50)).

3.4. Illustrative Examples

We close this section with three transformation examples. The first is a one dimensional truss discretized with a single three node quadratic element as depicted in Figure 3a. In this case, absolute shifted enrichment is used at all nodes and the transformation is given by

$$\begin{bmatrix} u_1 \\ u_1^{|s|} \\ u_2 \\ u_2^{|s|} \\ u_3 \\ u_3^{|s|} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} u_1^{\mathrm{a}} \\ u_1^{\mathrm{b}} \\ u_2^{\mathrm{a}} \\ u_2^{\mathrm{b}} \\ u_3^{\mathrm{b}} \\ u_3^{\mathrm{a}} \end{bmatrix} . \tag{52}$$

Each 2 × 2 sub-matrix on the block diagonal is identical to those in Eq. (51). This means that the presence of quadratic basis functions does not change the transformation structure. As mentioned earlier, a particular vector ordering is assumed in the transformed space to obtain this constant block diagonal structure. Specifically, at some nodes u_I^{a} appears before u_I^{b} while this is reversed at other nodes. If one needed to actually determine this ordering, then it would be necessary to evaluate nodal positions relative to the discontinuity. However, this information is not required during the transformation as the algebraic multigrid procedure does not make use of the specific ordering. Thus, there is no need to maintain any information about nodal positions within the transformation implementation.

The second example is from two dimensional linear elasticity discretized by a single triangular element using linear basis functions. The triangle is intersected by a crack as depicted in Figure 3b and so each of the three nodes has two standard basis functions corresponding to x- and y-displacements as well as two absolute shifted enriched basis functions corresponding to x- and y-displacements. The transformation is defined by

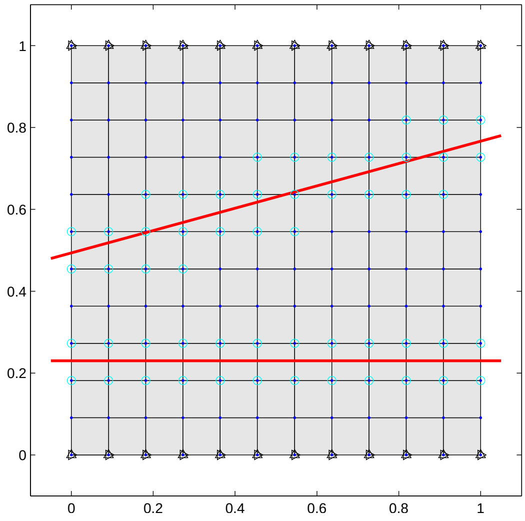


Figure 4. Mesh sub-divided by two cracks.

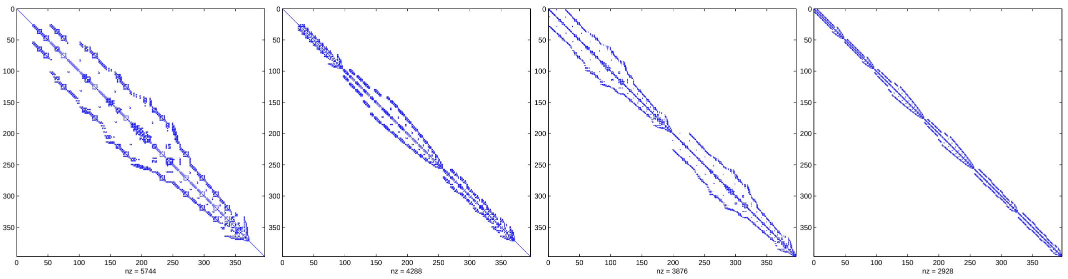
$$
\begin{bmatrix} u_{1,x} \\ u_{1,y} \\ u^{|s|}_{1,x} \\ u^{|s|}_{1,y} \\ u_{2,x} \\ u_{2,y} \\ u^{|s|}_{2,x} \\ u^{|s|}_{2,y} \\ u_{3,x} \\ u_{3,y} \\ u^{|s|}_{3,x} \\ u^{|s|}_{3,y} \end{bmatrix}
=
\left[\begin{array}{cccc|cccc|cccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hdashline
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\
\hdashline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1
\end{array}\right]
\begin{bmatrix} u^{\mathrm{a}}_{1,x} \\ u^{\mathrm{a}}_{1,y} \\ u^{\mathrm{b}}_{1,x} \\ u^{\mathrm{b}}_{1,y} \\ u^{\mathrm{b}}_{2,x} \\ u^{\mathrm{b}}_{2,y} \\ u^{\mathrm{a}}_{2,x} \\ u^{\mathrm{a}}_{2,y} \\ u^{\mathrm{b}}_{3,x} \\ u^{\mathrm{b}}_{3,y} \\ u^{\mathrm{a}}_{3,x} \\ u^{\mathrm{a}}_{3,y} \end{bmatrix} . \qquad (53)
$$

This same transformation applies even if the crack is located differently within the element. The only difference (changes in the ordering of some transformed unknowns) does not impact the AMG library. That is, the crack location and orientation do not affect the basic block structure of the transformation.

The last example corresponds to a large finite element mesh where two cracks effectively divide the domain into three physically independent regions (see Figure 4). As expected, the transformed mass and stiffness matrices in Figure 5 have three independent blocks, while no block structure is apparent in the untransformed stiffness matrix. It should be noted that in all cases a re-ordering algorithm is applied [30] only to reveal any possible block structure to the human eye, such permutation is not needed in a multigrid computation.

Before closing this section, we note one subtle but important matrix graph issue within the AMG scheme (given by Eq. (16) and Eq. (17)). In particular, we stated that a single vertex represents unknown degrees-of-freedom at a single mesh node. While this would be true for a standard FEM discretization, the situation is a bit different for a phantom node approach. In particular, while u^{a}_I and u^{b}_I are located at the same mesh location in the one dimensional examples, they correspond to unconnected degrees-of-freedom on opposite sides of the crack. Thus, they are treated as two different vertices in the AMG graph representation. Similarly, in the two dimensional example, the two degrees-of-freedom corresponding to $u^{\mathrm{a}}_{I,x}$ and $u^{\mathrm{a}}_{I,y}$ are treated as one vertex while $u^{\mathrm{b}}_{I,x}$ and $u^{\mathrm{b}}_{I,y}$ are treated as a second vertex. Notice in Eq. (53) that x and y displacements corresponding to a

DOI: 10.1002/nme



(a) $\boldsymbol{K}$ (b) $\boldsymbol{G}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{G}$, reordered (c) $\boldsymbol{M}$ (d) $\boldsymbol{G}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{G}$, reordered

Figure 5. Sparsity pattern for stiffness and mass matrix (blue dots indicate non-zero matrix entries); the matrix entries are re-ordered using the symmetric reverse Cuthill-McKee permutation [30] to make the block structure visible. Note that the mass matrix has 6 independent blocks, since mass matrix entries for x- and y-displacement are also independent from each other.

single vertex are ordered consecutively (which is typically required of most AMG codes). Finally, it is worth pointing out that the number of degrees-of-freedom per vertex is also constant with this phantom node representation (which is often also a requirement of AMG codes). This is in contrast to a traditional XFEM discretization where the most natural assignment of degrees-of-freedom to vertices would not lead to a constant number of degrees-of-freedom per vertex.

4. APPLYING ALGEBRAIC MULTIGRID TO XFEM DISCRETIZATIONS

4.1. Overview

Multigrid algorithms are efficient techniques for solving large linear systems [17, 18]. Their rapid convergence hinges on an interplay between smoothing and coarse correction. Smoothing is a simple iteration focused on reducing oscillatory error. The coarse correction is the formation and projection of a residual equation onto a coarse space. The central premise is that error can be represented at a coarser resolution once it has been smoothed. The solution to the projected system is then approximated (often via a recursive multigrid invocation), interpolated, and finally added to the fine level iterate. The key is that basic iterative procedures are normally efficient at reducing oscillatory error when applied to elliptic problems. Algorithm 1 illustrates a multigrid V-cycle. $\boldsymbol{A}_\ell$ is the discretization matrix on level ℓ. $\boldsymbol{R}_\ell$ restricts residuals from level ℓ to level ℓ+1, and $\boldsymbol{P}_\ell$ prolongates from level ℓ+1 to ℓ. $\mathcal{S}_\ell^{\mathrm{pre}}()$ and $\mathcal{S}_\ell^{\mathrm{post}}()$ are smoothing procedures. Algebraic multigrid provides an automated framework where grid transfers are generated algebraically and the discrete equations are projected with a triple matrix product, $\boldsymbol{A}_{\ell+1} = \boldsymbol{R}_\ell \boldsymbol{A}_\ell \boldsymbol{P}_\ell$. For symmetric systems only a prolongator

Algorithm 1 Multigrid V-cycle to solve $\boldsymbol{A}_\ell \boldsymbol{u} = \boldsymbol{b}$.

```
function MGV($\boldsymbol{A}_\ell, \boldsymbol{u}, \boldsymbol{b}, \ell$) :
if $\ell \neq \ell_{\max}$ then
    $\boldsymbol{u} \leftarrow \mathcal{S}_\ell^{\mathrm{pre}}(\boldsymbol{A}_\ell, \boldsymbol{u}, \boldsymbol{b})$
    $\boldsymbol{r} \leftarrow \boldsymbol{b} - \boldsymbol{A}_\ell \boldsymbol{u}$
    $\boldsymbol{c} \leftarrow 0$
    $\boldsymbol{c} \leftarrow$ MGV($\boldsymbol{A}_{\ell+1}, \boldsymbol{c}, \boldsymbol{R}_\ell \boldsymbol{r}, \ell+1$)
    $\boldsymbol{u} \leftarrow \boldsymbol{u} + \boldsymbol{P}_\ell \boldsymbol{c}$
    $\boldsymbol{u} \leftarrow \mathcal{S}_\ell^{\mathrm{post}}(\boldsymbol{A}_\ell, \boldsymbol{u}, \boldsymbol{b})$
else
    $\boldsymbol{u} \leftarrow \boldsymbol{A}_\ell^{-1} \boldsymbol{b}$
```

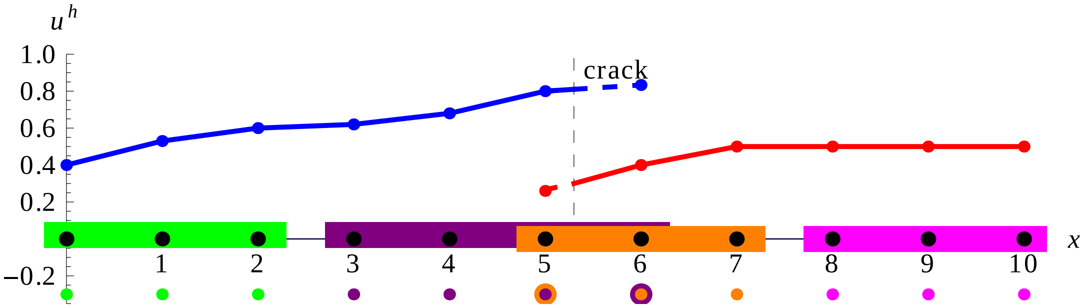


Figure 6. Exemplary 1d aggregation pattern (finest level): Same color implies same aggregate.

is developed as the restrictor is taken as the transpose of the prolongator to preserve symmetry of the AMG preconditioner.

In the following, we first describe the AMG process for the Heaviside enrichments along the crack face and, subsequently, for tip enrichments. Our intention is to leverage existing software and so we limit our discussion to AMG issues with XFEM implications and refer to [17, 18, 31, 32] for AMG details.

4.2. *Heaviside enrichments*

4.2.1. Aggregation and grid transfer operators For Heaviside enrichments, the main challenge is how to define the transfer operator $\boldsymbol{P}_\ell$. The process of creating a $\boldsymbol{P}_\ell$ begins by defining a matrix graph as described by Eq. (16) and Eq. (17). A coarsening strategy must then be chosen. In this paper, we focus on smoothed aggregation coarsening [32, 33], though this is not necessary for our XFEM/AMG approach. In the smoothed aggregation case, coarsening is accomplished by grouping the graph vertices into a set of *aggregates* $\mathcal{A}^i_\ell$, such that

$$\bigcup_{i=1}^{N_{\ell+1}} \mathcal{A}^i_\ell = \{\mathcal{V}_1, ..., \mathcal{V}_{N_\ell}\} \ , \ \mathcal{A}^i_\ell \cap \mathcal{A}^j_\ell = \emptyset \ , \ 1 \leq i < j \leq N_{\ell+1} \ , \tag{54}$$

where N_ℓ denotes the number of graph vertices on level ℓ and $N_{\ell+1}$ is the number of aggregates created during the aggregation phase. As each aggregate on level ℓ gives rise to one vertex on the next level, the matrix graph for level $\ell + 1$ has $N_{\ell+1}$ vertices. Aggregation specifics are not important here. The main idea is that each aggregate is essentially defined by first choosing an unaggregated vertex and grouping it with all of its unaggregated neighbor vertices.[‡] In this way, aggregates are comprised of neighboring vertices.

If we now consider this aggregation phase in the context of the phantom node approach, we recall that vertices within different disconnected regions correspond to solutions associated with opposite sides of a discontinuity. Clearly, two vertices from two different disconnected regions are not neighbors in the corresponding matrix graph. This means that vertices associated with opposite sides a crack will not be grouped into the same aggregate during the aggregation phase. This is depicted in Figure 6 for a one dimensional example. The blue and red lines denote some sample displacements to the left and to the right of the discontinuity respectively. Here, it is important to recall that at node five and at node six there are two vertices corresponding to $u^{\mathrm{a}}_5, u^{\mathrm{b}}_5, u^{\mathrm{a}}_6$, and u^{b}_6. Individual aggregates are depicted by rectangles and circles just below the rectangles. Two circles are given at node five and at node six. The inner circles correspond to u^{a}_5 and u^{b}_6 while the outer circles are u^{b}_5 and u^{a}_6. That is, outer circles depict vertices associated with *phantom* basis functions. These displacements correspond to the solution on the opposite side of the discontinuity compared to where the node is located. Thus, in this particular example the purple aggregate corresponds to

[‡] Aggregation schemes include heuristics allowing a couple of vertices which are neighbors of the neighbors to be assigned to the aggregate. These heuristics are oriented toward encouraging fairly equisized aggregates.

DOI: 10.1002/nme

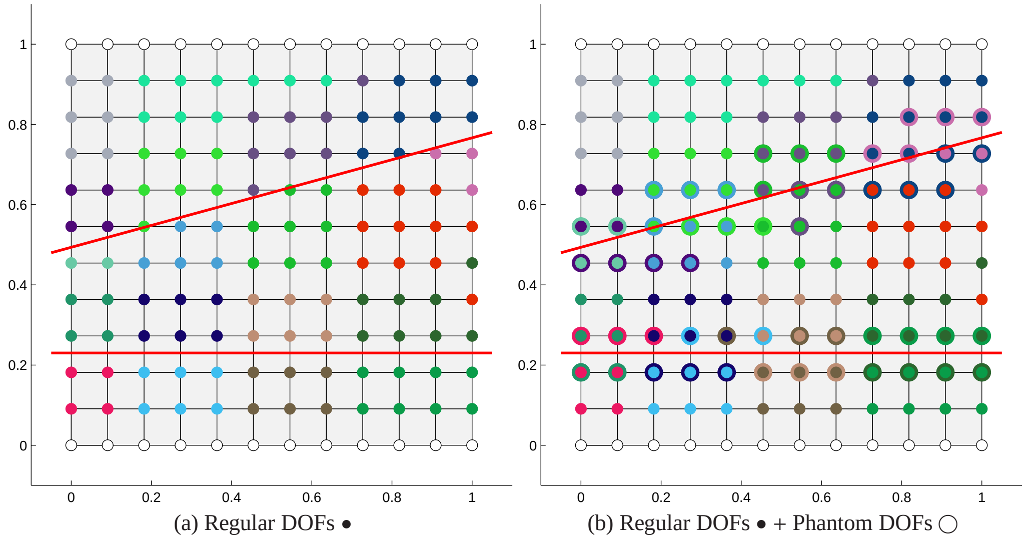


Figure 7. Exemplary 2D aggregation pattern (finest level): Same color implies same aggregate, white color indicates Dirichlet BC (not part of any aggregate).

the vertices associated with $u_3, u_4, u_5^{\mathrm{a}}$ and u_6^{a}, all of which are associated with displacements within Ω^{a}. Figure 7 depicts similar information in two dimensions.

With aggregates defined, a simple aggregate-wise prolongator can now be constructed,

$$\hat{\boldsymbol{P}}_\ell = \begin{pmatrix} \hat{\boldsymbol{P}}_\ell^{1,1} & & & \\ & \hat{\boldsymbol{P}}_\ell^{2,2} & & \\ & & \ddots & \\ & & & \hat{\boldsymbol{P}}_\ell^{N_{\ell+1},N_{\ell+1}} \end{pmatrix}, \tag{55}$$

where for simplicity of exposition we have assumed that all matrix rows associated with a specific aggregate are ordered consecutively. The detailed contents of the $\hat{\boldsymbol{P}}_\ell^{i,i}$ are unimportant for this paper. The key point is that basis functions do not span multiple aggregates. As the aggregates respect the discontinuity (for a phantom node discretization), then the prolongator is able to capture this discontinuity. This implies that if $\boldsymbol{A}_\ell$ has disconnected regions, then $\boldsymbol{A}_{\ell+1} = \hat{\boldsymbol{P}}_\ell^{\mathrm{T}} \boldsymbol{A}_\ell \hat{\boldsymbol{P}}_\ell$ will also have disconnected regions and so the coarse level discretization is guaranteed to properly represent the disconnected nature of a crack.

By contrast, an original XFEM (or shifted enrichment, or absolute shifted enrichment) version of $\boldsymbol{A}_\ell$ does not have disconnected regions. This means that the matrix graph contains edges whose two vertices are on opposite sides of cracks. Thus, aggregates generally cross discontinuities and so the corresponding prolongator will include interpolation basis functions which also cross cracks. As the algebraic multigrid method has no explicit information about cracks nor was it designed with XFEM discretizations in mind, it is most likely that the resulting coarse level discretization will not properly capture a strong discontinuity. That is, coarse solutions will be smooth across the crack and not accurately represent the physical features of the fine level discretization. Thus, it should not be surprising that the black-box application of AMG methods to original (or shifted enrichment, or absolute shifted enrichment) XFEM discretizations might be problematic while at the same time its application to a phantom node representation might be more promising.

In practice, the interpolation given by Eq. (55) is referred to as the tentative prolongator in smoothed aggregation terminology. It is often improved via

$$\boldsymbol{P}_\ell = (\boldsymbol{I} - \omega \boldsymbol{D}_\ell^{-1} \boldsymbol{A}_\ell) \hat{\boldsymbol{P}}_\ell \tag{56}$$

where the matrix $\boldsymbol{D}_\ell$ contains the diagonal entries of $\boldsymbol{A}_\ell$, and ω is an automatically generated parameter chosen so that oscillatory modes are damped. The details are again not so important in understanding the XFEM ramifications. What is important in our context is to recognize that Eq. (56) still respects discontinuities when $\boldsymbol{A}_\ell$ is a phantom node discretization. In particular, if $\hat{\boldsymbol{P}}_\ell$ does not include interpolation basis functions which traverse a discontinuity and if $\boldsymbol{A}_\ell$ has disconnected regions associated with cracks, then $\boldsymbol{P}_\ell$ will also not have basis functions crossing discontinuities. Thus, $\boldsymbol{A}_{\ell+1} = \boldsymbol{P}_\ell^{\mathrm{T}} \boldsymbol{A}_\ell \boldsymbol{P}_\ell$ will once again contain disconnected regions and is guaranteed to properly represent the disconnected nature of the discontinuities.

4.2.2. Null space In smoothed aggregation, users are required to supply something referred to as the near null space. For linear elasticity, this near null space corresponds to the rigid body modes. Thus, in two dimensions, the user should provide three vectors: one corresponding to a constant x direction translation, one corresponding to a constant y direction translation, and one corresponding to a constant rotation in the x-y plane. For example, the null space vectors for a 2D finite element problem with an assumed DOF ordering of

$$\boldsymbol{u} = \begin{bmatrix} u_{1,x} & u_{1,y} & \dots & u_{I,x} & u_{I,y} & \dots & u_{N,x} & u_{N,y} \end{bmatrix}^{\mathrm{T}} \tag{57}$$

is given as

$$\boldsymbol{N} = \begin{bmatrix} \boldsymbol{u}^x \\ \boldsymbol{u}^y \\ \boldsymbol{u}^{\mathrm{rot}} \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} 1 & 0 & \dots & 1 & 0 & \dots & 1 & 0 \\ 0 & 1 & \dots & 0 & 1 & \dots & 0 & 1 \\ x_1 & -y_1 & \dots & x_I & -y_I & \dots & x_N & -y_N \end{bmatrix}^{\mathrm{T}} \tag{58}$$

where x_I and y_I describe the position of node I. Thus, the last row describes a rigid body rotation around the origin. In 3D, users provide six vectors (three translations and three rotations).

For fracture problems, rigid body motion implies that the crack does not open or close during such movement. Hence, in the phantom approach, these vectors are naturally defined at all nodes including phantom nodes identical to how one would do for a standard FEM discretization. With original, shifted enrichment, or absolute shifted enrichment XFEM, the additional degrees of freedom alone describe the width of the crack opening. In other words, the enriched degrees of freedom do not contribute to rigid body modes of the full domain. Hence, one defines a *standard* near null space for degrees-of-freedom associated with unenriched basis functions and sets vector components associated with enriched degrees-of-freedom to zero. Thus, if node I has both unenriched and enriched degrees of freedom,

$$\boldsymbol{u} = \begin{bmatrix} u_{1,x} & u_{1,y} & \dots & u_{I,x} & u_{I,y} & u^{\mathrm{s}}_{I,x} & u^{\mathrm{s}}_{I,y} & \dots & u_{N,x} & u_{N,y} \end{bmatrix}^{\mathrm{T}}, \tag{59}$$

then the corresponding null space would contain 0 for the enriched degrees of freedom:

$$\boldsymbol{N} = \begin{bmatrix} 1 & 0 & \dots & 1 & 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 1 & \dots & 0 & 1 & 0 & 0 & \dots & 0 & 1 \\ x_1 & -y_1 & \dots & x_I & -y_I & 0 & 0 & \dots & x_N & -y_N \end{bmatrix}^{\mathrm{T}}. \tag{60}$$

When one applies the transformation given by Eq. (29) to the matrix, it is also necessary to apply $\boldsymbol{G}^{-1}$ to the near null space vectors before providing these to an AMG setup procedure

$$\boldsymbol{N}^* = \boldsymbol{G}^{-1} \boldsymbol{N}. \tag{61}$$

In this way, null space vectors automatically correspond to a proper phantom node representation.

4.2.3. Implementation The simplest implementation would explicitly form $\boldsymbol{G}$, $\boldsymbol{G}^{\mathrm{T}}$, $\boldsymbol{G}^{-1}$, and $\boldsymbol{G}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{G}$. The AMG setup phase would then be supplied $\boldsymbol{G}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{G}$ and the transformed null space. During the solve phase, one would apply the outer iterative solver to the original system, $\boldsymbol{A}$. Within the preconditioner, however, residual vectors would be transformed via $\boldsymbol{G}^{\mathrm{T}}$ before being passed to the

DOI: 10.1002/nme

multigrid V-cycle procedure. The approximation that results from this multigrid procedure would then be transformed back to the original space via $\boldsymbol{G}$ before being passed back to the outer iterative procedure. In this way, very few changes need to be made to either the XFEM or the AMG software. Of course, this simple implementation requires the storage of several additional matrices. It is, however, important to recognize that some storage can be avoided due to the the simple form of the transformation. In particular, it was shown in the previous section that it is generally possible to define $\boldsymbol{G}$ with the help of one 2×2 matrix which appears in several places along the block diagonal. The same obviously follows for $\boldsymbol{G}^{\mathrm{T}}$ and $\boldsymbol{G}^{-1}$. Thus, it is not strictly necessary to explicitly store these so long as one is able to perform matrix-vector and matrix-matrix products. For example, within the ML multigrid package [34], matrices can be specified by supplying only two functions: one which is able to apply matrix-vector products and the other which can supply a single matrix row 'on the fly' when it is requested. In this way, explicit storage of $\boldsymbol{G}$, $\boldsymbol{G}^{\mathrm{T}}$, $\boldsymbol{G}^{-1}$ can be avoided if one implements these implicitly. In fact, it is even possible to avoid the explicit storage of $\boldsymbol{G}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{G}$, though this is considerably more challenging from an implementation perspective. Of course, implicit representations may incur some additional cost when these operators are employed. While we have not implemented storage-saving measures, our best guess it that efficient implementations would add only modestly to the setup phase run time and even less to the solve phase run time.

4.3. *Tip Enrichments*

4.3.1. Null space, transformation, and transfer operators Tip enrichments describe a local feature around the crack tip. In the original XFEM publications, these tip enrichments only span the element that contains the crack tip, while more recently, elements in a given radius around the crack tip are enriched [35]. From an AMG perspective, tips are considered as high frequency features that are best treated by smoothing, because they cannot easily be represented on a coarser grid of a multigrid hierarchy. Hence, our strategy is to *not* include the extra degrees into the transfer operators. To do this without modifying 'AMG software', we exploit the way that smoothed aggregation constructs prolongators. Without going into details, the prolongator construction usually ignores degrees of freedom that do not contribute to the null space. In other words, if a degree of freedom is not needed to represent near null space modes, they are not interpolated from coarser grids. Hence, in practice, we only need to make sure that the null space vector has zero entries for the enriched DOFs. As described in Section 4.2.2, the null space vector automatically contains zeros for enriched DOFs when shifted enrichment, absolute shifted enrichment, or original XFEM form is used. To preserve this quality, an identity matrix is used for tip enriched DOFs in the transformation matrix $\boldsymbol{G}$. This means that all Heaviside DOFs are in phantom node form while all tip enriched DOFs are in their original form after the transformation. Thus, the null space associated with tip DOFs are all zero and so they are ignored within the prolongator.

4.3.2. Smoothing As mentioned in the introduction, tip enrichment functions also introduce a significant level of ill-conditioning. As a remedy, we propose a multiplicative Schwarz solution strategy for the finest level, where, in addition to a smoothing step for the entire domain, we solve directly for all unknowns that are attached to nodes with tip enriched elements. If multiple disconnected regions of unknowns are present, each of them can be solved independently. Formally, let set $\mathcal{T}$ contain all DOFs associated with the tip enrichment F plus the normal degree-of-freedom associated with tip elements. For example, this would result in a block size of 40×40 (8 standard DOFs + 32 enriched DOFs) for a 2D bilinear 4 node quadrilateral element, as used in the examples. Now define a smoother $\mathcal{S}^{\mathrm{tip}}$ that acts only on $\mathcal{T}$ as

$$\mathcal{S}^{\mathrm{tip}}_{\mathcal{T}\mathcal{T}} = \tilde{\boldsymbol{A}}^{-1}_{\mathcal{T}\mathcal{T}} \tag{62}$$

$$\mathcal{S}^{\mathrm{tip}}_{ij} = 0 \text{ if } i \notin \mathcal{T} \text{ or } j \notin \mathcal{T} \tag{63}$$

where $\tilde{\boldsymbol{A}} = \boldsymbol{G}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{G}$ denotes the transformed system. Note that the meaning of $\mathcal{T}$ does not change during the application of $\boldsymbol{G}$, because tip enriched DOFs and unenriched DOFs at tip nodes are not altered during the transformation.

With this definition, the two-step (multiplicative Schwarz) pre-smoother on the finest level is

$$\boldsymbol{u} \leftarrow \text{GaussSeidel}(\boldsymbol{u}, \tilde{\boldsymbol{A}}, \boldsymbol{b}) \tag{64}$$
$$\boldsymbol{u} \leftarrow \boldsymbol{u} + \mathcal{S}^{\text{tip}}(\boldsymbol{b} - \tilde{\boldsymbol{A}}\boldsymbol{u}) \tag{65}$$

and the post-smoother is defined as

$$\boldsymbol{u} \leftarrow \boldsymbol{u} + \mathcal{S}^{\text{tip}}(\boldsymbol{b} - \tilde{\boldsymbol{A}}\boldsymbol{u}) \tag{66}$$
$$\boldsymbol{u} \leftarrow \text{GaussSeidel}(\boldsymbol{u}, \tilde{\boldsymbol{A}}, \boldsymbol{b}) \ . \tag{67}$$

The inverted order of the smoothing is necessary to obtain a symmetric operator.

For 3D problems, the block size is expected to be larger, because the tip enrichments would be placed along a line, hence several elements with tips are connected. Nevertheless, the block size should stay within reasonable limits such that the proposed direct solve is not a limiting factor.

4.4. *Summary*

Summarizing all algorithm components needed to use an existing smoothed aggregation library, the user must provide the following input:

1. construct null space with standard rigid body modes for unenriched DOFs, zero contributions for tip DOFs, and for Heaviside DOFs one needs either zero contributions for shifted and original XFEM or standard contributions for phantom node representations,
2. establish transformation matrix and transform linear system before passing it to the AMG library,
3. ensure pre-dropping of weak graph connections, which is part of most existing AMG libraries, to remove artifacts (near-zeros) stemming from the $\boldsymbol{G}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{G}$ transformation, and
4. employ additional smoothing on finest grid for elements with tip DOFs.

5. EXAMPLES

We concentrate on static problems only as they are usually harder to solve than system matrices representing dynamic systems. Dynamic problems can be solved in exactly the same fashion.

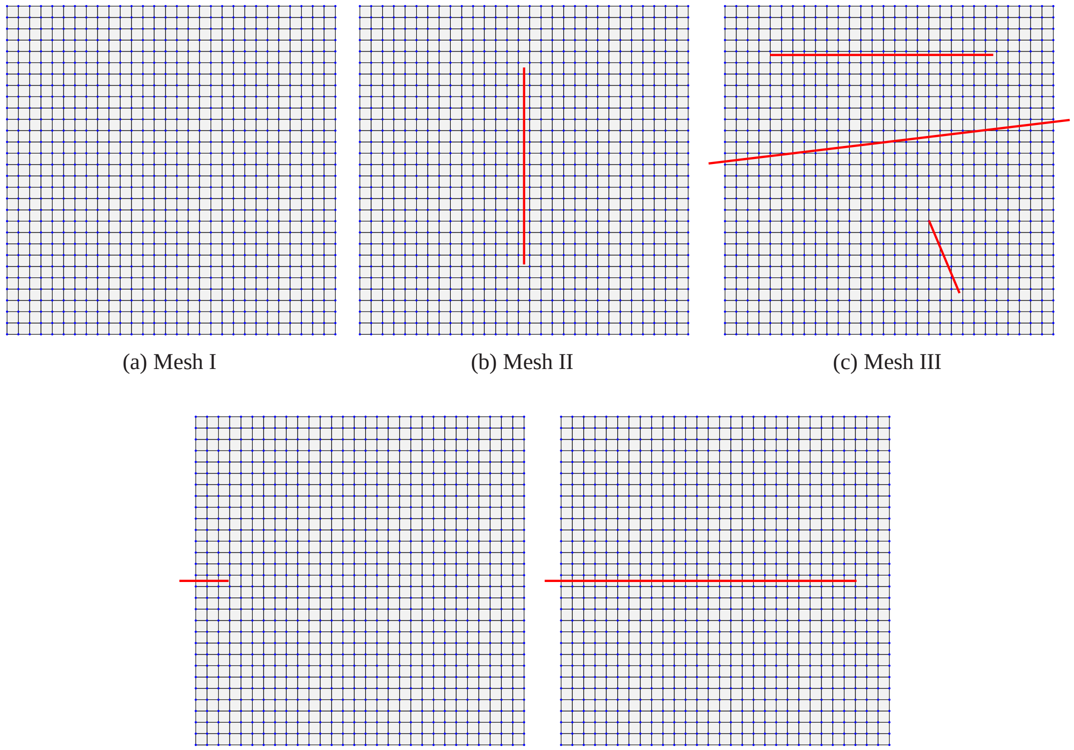
5.1. *XFEM with only jump enrichments*

We first focus on somewhat arbitrary crack locations and then take a closer look at the crack location within an element.

5.1.1. General problems Figure 8 illustrates a range of examples. The domain is of size 1 × 1 with zero Dirichlet boundary conditions at the top and bottom surface, zero Neumann conditions on the left surface, and a constant-load Neumann condition on the right surface. Note that in Figure 8c the crack crosses the entire domain effectively creating two independent sub-domains. For comparison, we also include an FEM example without any cracks in Figure 8a.

The results are given in Table II. The iteration numbers use a block-Gauss-Seidel smoother with only 1 level (basically a standard iterative solver) to the original shifted enrichment system and to the transformed system (columns 1 and 3) and use the proposed multilevel solver on the original and transformed system (columns 2 and 4). A dash indicates that convergence is not achieved within 200 iterations. Applying the AMG solver to the transformed system gives iteration counts that are *relatively* independent of the mesh refinement. Although there is some growth in iteration numbers, the importance of applying the transformation in conjunction with AMG is clearly apparent.

5.1.2. Sensitivity to crack distance from element nodes Locating the crack surface closer to an edge or node generates enriched approximations that are either becoming linear dependent to the regular



(a) Mesh I

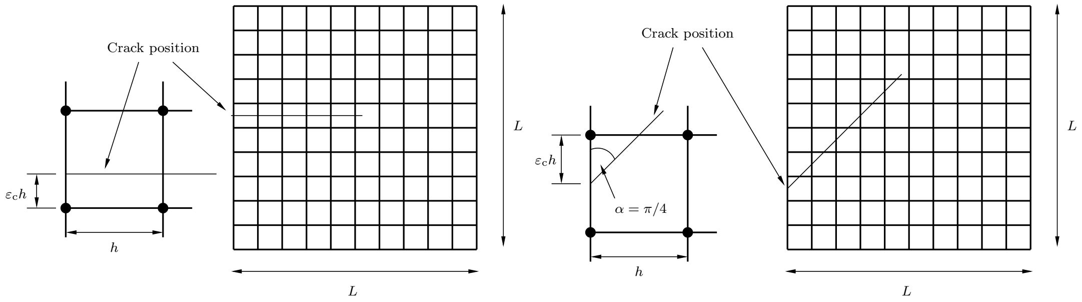
(b) Mesh II

(c) Mesh III

(d) Mesh 1a

(e) Mesh 1c

Figure 8. Illustrative examples.



(a) horizontal crack

(b) 45°-crack

Figure 9. Setup for sensitivity analysis.

finite element approximation or, in the case of the phantom node approach leads to a DOF with almost zero support. As a result, the condition number of the linear system can be arbitrarily higher than for a comparable FEM problem without cracks. These high condition numbers are well known and have been reported in a number of studies, e.g. [2, 35]. The general solution is to not enrich such nodes anymore [2] to avoid a rank deficient system matrix. Two studies are depicted in Figure 9a and Figure 9b, where a $L \times L$ domain is intersected by a horizontal and a 45° crack, respectively. In the horizontal case, the crack end points are $(0.0,\ \ 0.5L + \varepsilon_c h)$ and $(0.5L - 0.25h,\ \ 0.5L + \varepsilon_c h)$. For the 45° crack, the end points are $(0.0,\ \ (3 - \varepsilon_c)h)$ and $(0.5L - 0.25h,\ \ 0.5L + (2.75 - \varepsilon_c)h)$. As ε_c approaches 0, the system becomes more ill-conditioned. Results are given in Table III and

Table II. Iteration and condition numbers for the example problems depicted in Figure 8 with only Heaviside enrichments enabled.

Case	$n_e \times n_e$	$\alpha_{\text{cond.}}$	n_{iter}			
			$\boldsymbol{A}$		$\boldsymbol{G}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{G}$	
			1L	ML	1L	ML
I	30×30	3e+03	32	9	32	9
	60×60	1e+04	63	10	63	10
	90×90	3e+04	93	11	93	11
	120×120	5e+04	123	11	123	11
II	30×30	1e+04	46	23	43	11
	60×60	5e+04	86	34	84	12
	90×90	1e+05	127	42	126	15
	120×120	2e+05	170	49	167	15
III	30×30	6e+05	59	40	53	11
	60×60	3e+06	109	55	101	11
	90×90	8e+06	156	68	153	12
	120×120	8e+06	-	80	-	14
1a	30×30	1e+05	54	16	54	11
	60×60	4e+05	106	20	105	13
	90×90	1e+06	157	24	157	16
	120×120	2e+06	-	26	-	16
1c	30×30	2e+07	78	38	76	16
	60×60	7e+07	150	52	146	16
	90×90	1e+08	-	63	-	18
	120×120	2e+08	-	73	-	18

Table IV. Note that by changing h and ε_{c}, different physical problems are generated, which makes iteration counts for varying h formally not directly comparable. Nevertheless, it can be seen that for a given h, the iterative performance for both the 1-level method and the multigrid solution is almost independent with respect to ε_{c}. The reason for this is that Gauss-Seidel is generally insensitive to conditioning problems that could be resolved by a simple diagonal scaling of the linear system.

For ε_{c} smaller than the given results the AMG process breaks down due to numerical precision limits. As a remedy, not enriching unknowns with such small support as proposed e.g. in [2] makes the system solvable again. In [2], a node was not enriched, if the support on one side of the crack divided by the element area was below a threshold of $1 \cdot 10^{-4}$. For our 45°-crack example, the area ratio $A_{\mathrm{UL}}/A_{\mathrm{ele}}$ – the upper left triangle area A_{UL} divided by the element area A_{ele} – can be computed as $A_{\mathrm{UL}}/A_{\mathrm{ele}} = \varepsilon_{\mathrm{c}}^2/2$. Choosing $\varepsilon_{\mathrm{c}} = 0.001$ results in an area ratio of $5 \cdot 10^{-7}$, which is about two magnitudes smaller than the limit given [2]. For the horizontal crack, ε_{c} directly gives the area fraction. Hence, we conclude that the proposed multigrid scheme is able to cover all crack configurations that occur in practical XFEM simulation, if above threshold for enriching nodes is used.

5.2. *XFEM with jump and tip enrichments enrichments*

We now add tip enrichments around crack tips (in addition to the Heaviside enrichment) and focus on the transformed system. Comparison of Table V with Table II clearly shows that for the same crack position, the tip enrichment makes the linear system more difficult to solve. First of all, the condition number is always higher, if tip enriched DOFs are included. Correspondingly, the iteration counts in column 1 (no multigrid) are always higher than in Table V. Applying AMG to the

Table III. Results for crack location sensitivity, horizontal crack, Heaviside enrichment only

		$\boldsymbol{A}$			$\boldsymbol{G}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{G}$		
		n			n		
ε_{c}	$n_e \times n_e$	1L	ML	α_{cond}	1L	ML	α_{cond}
0.5	10×10	26	14	2e+04	25	9	1e+04
	20×20	47	20	6e+04	44	13	4e+04
	40×40	88	25	2e+05	86	13	2e+05
	80×80	169	43	8e+05	167	17	7e+05
0.01	10×10	27	15	2e+04	24	9	2e+04
	20×20	47	22	6e+04	45	13	5e+04
	40×40	88	26	2e+05	85	13	2e+05
	80×80	170	44	9e+05	168	17	7e+05
0.0001	10×10	27	15	2e+06	24	9	1e+06
	20×20	47	22	2e+06	45	13	1e+06
	40×40	88	26	2e+06	85	13	1e+06
	80×80	170	45	2e+06	168	17	2e+06
0.000001	10×10	27	15	2e+08	24	9	1e+08
	20×20	47	22	2e+08	45	13	1e+08
	40×40	88	26	2e+08	85	13	1e+08
	80×80	170	45	2e+08	168	17	1e+08
0.00000001	10×10	27	15	2e+10	24	9	1e+10
	20×20	47	22	2e+10	45	13	1e+10
	40×40	88	26	2e+10	85	13	1e+10
	80×80	170	45	2e+10	168	17	1e+10

Table IV. Results for crack location sensitivity, 45°-crack, Heaviside enrichment only

		$\boldsymbol{A}$			$\boldsymbol{G}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{G}$		
		n_{iter}			n_{iter}		
ε_{c}	$n_e \times n_e$	1L	ML	α_{cond}	1L	ML	α_{cond}
0.5	10×10	29	18	3e+04	27	10	2e+04
	20×20	52	25	1e+05	49	14	7e+04
	40×40	95	36	3e+05	92	14	3e+05
	80×80	182	50	1e+06	178	18	1e+06
0.1	10×10	31	20	2e+06	27	11	2e+06
	20×20	53	27	2e+06	48	14	2e+06
	40×40	96	37	3e+06	92	14	2e+06
	80×80	184	53	4e+06	178	16	3e+06
0.01	10×10	31	20	3e+10	27	11	2e+10
	20×20	53	27	3e+10	48	14	2e+10
	40×40	96	38	3e+10	91	14	2e+10
	80×80	183	53	3e+10	178	16	2e+10
0.001	10×10	31	21	3e+14	27	11	2e+14
	20×20	53	27	3e+14	48	14	2e+14
	40×40	97	38	3e+14	92	14	2e+14
	80×80	183	53	3e+14	178	16	2e+14

Table V. Heaviside and Tip Enrichment: Results for transformed system $\boldsymbol{G}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{G}$. The abbreviations are defined as: 1L: one level, ML: multilevel, ML, TS: multilevel with tip smoother on finest grid.

Case	$n_e \times n_e$	$\alpha_{\text{cond.}}$	n_{iter}		
			1L	ML	ML, TS
I	30×30	3e+03	32	9	9
	60×60	1e+04	63	10	10
	90×90	3e+04	93	11	11
	120×120	5e+04	123	11	11
II	30×30	2e+07	115	75	18
	60×60	8e+08	-	97	20
	90×90	8e+09	-	114	23
	120×120	3e+10	-	141	21
III	30×30	5e+07	143	94	18
	60×60	1e+09	-	158	20
	90×90	2e+10	-	-	20
	120×120	3e+10	-	-	24
1a	30×30	6e+05	66	31	16
	60×60	3e+06	117	30	16
	90×90	1e+07	165	33	20
	120×120	2e+07	-	32	19
1c	30×30	1e+08	86	34	20
	60×60	7e+08	157	34	21
	90×90	2e+09	-	35	24
	120×120	3e+09	-	34	23

transformed system (column 2) does improve the iteration count substantially. While for problem 1a and 1c this may already be a practically useful number of iterations, iteration counts for problem II and III are still not satisfactory. Only the multiplicative Schwarz smoothing gives an almost optimal number of iterations. More importantly, the difference between cases II and III versus 1a and 1c vanishes, such that all XFEM simulations converge in less than 30 iterations for the given mesh sizes.

Note that the position of the crack tip within the element does not significantly influence the condition number of the problem as the size of the associated matrix entries is not changed significantly (as in the case of Heaviside enrichments). Hence, no degeneration of the linear system is expected. For demonstration, we repeated the 45°-crack simulation (Figure 9b), this time with both Heaviside and tip enrichments enabled. It can be seen in Table VI that moving the crack tip within the element does not change the iteration count.

6. SUMMARY & CONCLUSION

We presented an algebraic multigrid approach to solve linear systems arising from applications where strong discontinuities are modeled by XFEM. In particular, we presented techniques that are applicable to fracture problems, which typically use Heaviside enrichment functions to model the crack face and special tip enrichment functions to model the near tip displacement and stress field.

We demonstrated, that the phantom node approach for modeling the crack face naturally fits the assumptions that underlay existing AMG methods for finite elements. For original and shifted XFEM approximations, we provided a simple and inexpensive – in terms of both memory and computation time – transformation to the phantom node representation such that software changes

Table VI. Results for crack location sensitivity, 45°-crack, Heaviside and tip enrichments

		$\boldsymbol{A}$			$\boldsymbol{G}^{\mathrm{T}}\boldsymbol{AG}$		
		n_{iter}			n_{iter}		
ε_{c}	$n_e \times n_e$	1L	ML, TS	α_{cond}	1L	ML, TS	α_{cond}
0.5	10×10	33	22	1e+05	30	17	1e+05
	20×20	53	28	3e+05	50	19	3e+05
	40×40	96	39	7e+05	93	21	6e+05
	80×80	182	55	2e+06	179	25	2e+06
0.1	10×10	34	24	3e+06	31	18	3e+06
	20×20	54	30	3e+06	50	20	3e+06
	40×40	96	41	3e+06	92	21	3e+06
	80×80	184	57	4e+06	179	25	3e+06
0.01	10×10	35	25	3e+10	31	18	3e+10
	20×20	55	30	3e+10	50	20	3e+10
	40×40	97	41	3e+10	92	21	2e+10
	80×80	184	58	3e+10	179	25	2e+10
0.001	10×10	35	25	3e+14	31	18	3e+14
	20×20	55	30	3e+14	50	20	3e+14
	40×40	97	41	3e+14	92	21	2e+14
	80×80	184	58	3e+14	179	25	2e+14

are not needed within either the XFEM or the multigrid libraries. Tip enrichments describe local features and so it is best not to include them into the grid transfer operators. Hence, such enrichments are smoothed on the finest level only as they are not present on coarser levels. (Block-)Gauss-Seidel is used for smoothing combined with a multiplicative Schwarz method on the finest level to address the ill-conditioning associated with tip degrees of freedom. This entails a direct solve for each set of unknowns connected to a tip element.

The examples demonstrate the effectiveness of the approach. The number of iterations scales well with the mesh size. In addition, the proposed method is insensitive to the crack position with respect to element edges and nodes. It should be possible to apply the same transformation ideas to branching cracks. Since branches can be modeled by the shifted or original XFEM as well as the phantom node approach, such a transformation could be established for 2D and 3D problems in a similar fashion as we demonstrated for non-branching cracks.

7. ACKNOWLEDGMENT

The authors would like to thank Badri Hiriyur and Haim Waisman for providing XFEM routines used to prepare the numerical examples in this paper.

REFERENCES

[1] Belytschko T, Black T. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering* 1999; **45**(5):601–620, doi: 10.1002/(SICI)1097-0207(19990620)45:5⟨601::AID-NME598⟩3.0.CO;2-S.

[2] Daux C, Moës N, Dolbow J, Sukumar N, Belytschko T. Arbitrary cracks and holes with the extended finite element method. *International Journal for Numerical Methods in Engineering*

2000; **48**(12):1741–1760, doi:10.1002/1097-0207(20000830)48:12⟨1741::AID-NME956⟩3.0.CO;2-L.

[3] Moës N, Dolbow J, Belytschko T. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering* 1999; **46**(1):131–150, doi:10.1002/(SICI)1097-0207(19990910)46:1⟨131::AID-NME726⟩3.0.CO;2-J.

[4] Stazi FL, Budyn E, Chessa J, Belytschko T. An extended finite element method with higher-order elements for curved cracks. *Computational Mechanics* May 2003; **31**(1):38–48, doi:10.1007/s00466-002-0391-2.

[5] Song JH, Areias PMA, Belytschko T. A method for dynamic crack and shear band propagation with phantom nodes. *Int. J. Numer. Meth. Engng.* 2006; **67**(6):868–893, doi:10.1002/nme.1652.

[6] Moës N, Gravouil A, Belytschko T. Non-planar 3-D crack growth by the extended finite element method and level sets, part i: Mechanical model. *International Journal for Numerical Methods in Engineering* 2002; **53**(11):2549–2568, doi:10.1002/nme.429.

[7] Sukumar N, Chopp DL, Moës N, Belytschko T. Modeling holes and inclusions by level sets in the extended finite-element method. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**(46-47):6183–6200, doi:10.1016/S0045-7825(01)00215-8.

[8] Groß S, Reusken A. An extended pressure finite element space for two-phase incompressible flows with surface tension. *Journal of Computational Physics* May 2007; **224**(1):40–58, doi:10.1016/j.jcp.2006.12.021.

[9] Legay A, Chessa J, Belytschko T. An Eulerian-Lagrangian method for fluid-structure interaction based on level sets. *Computer Methods in Applied Mechanics and Engineering* Mar 2006; **195**(17-18):2070–2087, doi:10.1016/j.cma.2005.02.025.

[10] Gerstenberger A, Wall WA. An extended finite element method / Lagrange multiplier based approach for fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering* Mar 2008; **197**(19-20):1699–1714, doi:10.1016/j.cma.2007.07.002.

[11] Mayer U, Popp A, Gerstenberger A, Wall W. 3d fluid-structure-contact interaction based on a combined XFEM FSI and dual mortar contact approach. *Computational Mechanics* Jun 2010; **46**(1):53–67, doi:10.1007/s00466-010-0486-0.

[12] Gerstenberger A. An XFEM based fixed-grid approach to fluid-structure interaction. PhD Thesis, Technische Universität München 2010. URL http://mediatum.ub.tum.de/node?id=973360.

[13] Zilian A, Legay A. The enriched space-time finite element method (EST) for simultaneous solution of fluid-structure interaction. *International Journal for Numerical Methods in Engineering* 2008; **75**(3):305–334, doi:10.1002/nme.2258.

[14] Chessa J, Belytschko T. Arbitrary discontinuities in space-time finite elements by level sets and X-FEM. *International Journal for Numerical Methods in Engineering* 2004; **61**(15):2595–2614, doi:10.1002/nme.1155.

[15] Fries TP, Belytschko T. The extended/generalized finite element method: An overview of the method and its applications. *Int. J. Numer. Meth. Engng.* 2010; **84**(3):253–304, doi:10.1002/nme.2914.

[16] Menk A, Bordas SPA. A robust preconditioning technique for the extended finite element method. *Int. J. Numer. Meth. Engng.* 2011; **85**(13):1609–1632, doi:10.1002/nme.3032.

[17] Briggs WL, Henson VE, McCormick S. *A Multigrid Tutorial, Second Edition*. SIAM: Philadelphia, 2000.

[18] Trottenberg U, Oosterlee C, Schüller A. *Multigrid*. Academic Press: London, 2001.

[19] Rannou J, Gravouil A, Combescure A. A multigrid extended finite element method for elastic crack growth simulation. *European J. of Comp. Mech.* 2007; **16**:161–182.

[20] Gravouil A, Rannou J, Baïetto MC. A local multi-grid x-fem approach for 3d fatigue crack growth. *International Journal of Material Forming* 2008; **1**:1103–1106, doi:10.1007/s12289-008-0212-z. 10.1007/s12289-008-0212-z.

[21] Cho D, Zikatanov L. A multilevel preconditioning for generalized finite element method problems on unstructured simplicial meshes. *Journal of Numerical Mathematics* 2007; **15**(3):163–180, doi:10.1515/jnma.2007.008.

[22] Xu J, Zikatanov LT. On multigrid methods for generalized finite element methods. *Meshfree methods for partial differential equations (Bonn, 2001)*, *Lect. Notes Comput. Sci. Eng.*, vol. 26. Springer: Berlin, 2003; 401–418.

[23] Berger-Vergiat L, Waisman H, Hiriyur B, Tuminaro R, Keyes D. Inexact schwarz-algebraic multigrid preconditioners for crack problems modeled by extended finite element methods. *Int. J. Numer. Meth. Engng* 2011; :n/a–n/adoi:10.1002/nme.3318.

[24] Hiriyur B, Tuminaro RS, Waisman H, Boman EG, Keyes DE. A quasi-algebraic multigrid approach to fracture problems based on the extended finite element method. *to appear in SIAM J. Sci. Comput.* 2012; .

[25] Belytschko T, Moës N, Usui S, Parimi C. Arbitrary discontinuities in finite elements. *International Journal for Numerical Methods in Engineering* 2001; **50**(4):993–1013, doi:10.1002/1097-0207(20010210)50:4⟨993::AID-NME164⟩3.0.CO;2-M.

[26] Hughes T. *The Finite Element Method – Linear Statics and Dynamic Finite Element Analysis*. Prentice Hall, 1987.

[27] Hansbo A, Hansbo P. A finite element method for the simulation of strong and weak discontinuities in solid mechanics. *Computer Methods in Applied Mechanics and Engineering* Aug 2004; **193**(33-35):3523–3540, doi:10.1016/j.cma.2003.12.041.

[28] Areias PMA, Belytschko T. Analysis of three-dimensional crack initiation and propagation using the extended finite element method. *Int. J. Numer. Meth. Engng.* 2005; **63**(5):760–788, doi:10.1002/nme.1305.

[29] Areias PM, Belytschko T. A comment on the article ”a finite element method for simulation of strong and weak discontinuities in solid mechanics” by a. hansbo and p. hansbo [comput. methods appl. mech. engrg. 193 (2004) 3523–3540]. *Computer Methods in Applied Mechanics and Engineering* Feb 2006; **195**(9-12):1275–1276, doi:10.1016/j.cma.2005.03.006.

[30] Cuthill E, McKee J. Reducing the bandwidth of sparse symmetric matrices. *Proc. 24th Nat. Conf. ACM*, 1969; 157–172.

[31] Ruge J, Stüben K. Algebraic multigrid (AMG). *Multigrid Methods*, *Frontiers in Applied Mathematics*, vol. 3, McCormick SF (ed.). SIAM: Philadelphia, PA, 1987; 73–130.

[32] Vanek P, Mandel J, Brezina M. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing* 1996; **56**:179–196.

[33] Vaněk P, Brezina M, Mandel J. Convergence of algebraic multigrid based on smoothed aggregation. *Numerische Mathematik* 2001; **88**:559–579.

[34] Gee M, Siefert C, Hu J, Tuminaro R, Sala M. ML 5.0 smoothed aggregation user's guide. *Technical Report SAND2006-2649*, Sandia National Laboratories, Albuquerque, NM, 87185 2006.

[35] Béchet E, Minnebo H, Moës N, Burgardt B. Improved implementation and robustness study of the X-FEM for stress analysis around cracks. *International Journal for Numerical Methods in Engineering* 2005; **64**(8):1033–1056, doi:10.1002/nme.1386.